AD-A037 117   MITRE CORP BEDFORD MASS   F/G 9/2

COMPUTER PERFORMANCE EVALUATION DURING SYSTEM ACQUISITION.(U)

JAN 77 B A FELDMEYER   F19628-77-C-0001

UNCLASSIFIED   MTR-3290   ESD-TR-76-358   NL

1 OF 1

AD A037117

END

DATE FILMED

4-77

ESD-TR-76-358 MTR-3290

COMPUTER PERFORMANCE EVALUATION DURING SYSTEM ACQUISITION

JANUARY 1977

Prepared for

DEPUTY FOR COMMAND & MANAGEMENT SYSTEMS

ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
Hanscom Air Force Base, Bedford, Massachusetts

Project No. 572C
Prepared by
THE MITRE CORPORATION
Bedford, Massachusetts
Contract No. F19628-77-C-0001

REVIEW AND APPROVAL

This technical report has been reviewed and is approved for publication.

ANN MELANSON
Techniques Engineering Division

JOHN T. HOLLAND, Lt Colonel, USAF
Chief, Techniques Engineering Div.

FOR THE COMMANDER

FRANK J. EMMA, Colonel, USAF
Director, Computer Systems Engineering
Deputy for Command & Management Systems

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19 REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 18 ESD TR-76-358 / 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): 6 COMPUTER PERFORMANCE EVALUATION DURING SYSTEM ACQUISITION.	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER: 14 MTR-3290
7. AUTHOR(s): 10 Bruce A. Feldmeyer	8. CONTRACT OR GRANT NUMBER(s): 15 F19628-77-C-0001
9. PERFORMING ORGANIZATION NAME AND ADDRESS: The MITRE Corporation, Box 208, Bedford, MA 01730	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project No. 572C
11. CONTROLLING OFFICE NAME AND ADDRESS: Deputy for Command & Management Systems, Electronic Systems Division, AFSC, Hanscom Air Force Base, Bedford, MA 01731	12. REPORT DATE: 11 JAN[redacted]77
	13. NUMBER OF PAGES: 31
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 12 32p.	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

9 Technical rept.,

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

COMPUTER PERFORMANCE EVALUATION
SYSTEM ACQUISITION
SYSTEM INSTRUMENTATION
SYSTEM TESTING

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report discusses the applications of various computer performance evaluation tools and techniques during the acquisition of Air Force C^3 systems. It is concluded that the tool with the broadest range of applications is computer simulation, and that the most practical approach is to develop a series of models rather than a single model which evolves with the system through the conceptual, validation and full-scale development phases.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

235 050

mt

ACKNOWLEDGEMENT

This report has been prepared by the MITRE Corporation under Project 572C. The contract is sponsored by the Electronic Systems Division, Air Force Systems Command, Hanscom Air Force Base, Massachusetts.

TABLE OF CONTENTS

TABLE OF CONTENTS
(con't)

LIST OF TABLES

SECTION I

INTRODUCTION

C-CUBED

PURPOSE AND SCOPE

This report discusses the applications of various computer performance evaluation (CPE) tools and techniques such as hardware monitors, software monitors, and computer simulation during the acquisition of Air Force Command, Control and Communications (C^3) systems. It is intended primarily for analysts concerned with the specification, measurement and evaluation of the performance of the computer resources in these systems. It can be used to augment the software acquisition guidebooks (e.g., Software Verification) and checklists being developed to assist in the review of system acquisition documents.

This report does not discuss either performance evaluation per se or the activities, milestones or products during system acquisition. It is assumed that the reader is familiar with these subjects. (References 1 and 2.)

APPROACH

The basic approach taken in this study was to investigate the past and current uses of CPE tools and techniques in selected C^3 systems. Major efforts were directed at case studies of the Tactical Air Control Center (TACC) Improvement Program of Project 485L and the NORAD Cheyenne Mountain Complex Improvement Program (Project 427M). These case studies were supplemented by limited investigations of the performance activities of the Military Airlift Command Integrated Management System (MACIMS) and the Airborne Warning and Control System (AWACS).

This report is based on the findings of these case studies. The report is organized by acquisition phase, with three main sections on the conceptual, validation and full-scale development phases. These sections are preceded by a technical overview section and followed by a summary section.

Because of the tremendous complexity of C^3 systems, their life cycles vary significantly. This report describes performance activities as taking place in their most typical acquisition phase. However, a given system would be expected to differ somewhat from the typical. In fact on many C^3 programs there has not been a major distinction made between the conceptual and validation phases, and

for one-of-a-kind systems, many validation phase activities take place during full-scale development.

SECTION II

TECHNICAL OVERVIEW

This section contains a brief overview of those aspects of computer workloads and performance characteristics that do not apply to a specific acquisition phase. The basic terminology used in the remainder of the report is developed and considerations that apply equally to the conceptual, validation and full-scale development phases are stated. Workload and performance characteristic considerations which normally take place during a specific phase are contained in the appropriate section.

WORKLOAD

The workload for a system can be represented by three different levels: modal, peak and minimal. The modal workload represents the workload the system will be processing most of the time. The peak workload represents the system load during periods of heavy activity. As such it is the peak workload, not the modal workload, that sizes the system. In the C^3 environment it is also common to have the most stringent response time requirements occur during the peak workload period. The third type of workload is the minimal workload which represents the processing requirements during a degraded mode of operation (when not all system components are operational). This workload normally represents the high priority, operationally required functions that are critical to the system and is of primary concern when determining back-up requirements.

Although the modal workload is not normally the primary consideration in either sizing the system or determining back-up requirements, it may still be important. This is especially true if there are significant differences between the types of processing done during periods of average and peak periods. For example, if the modal workload consists of producing several historical management reports that are not produced during peak workload periods, the average workload may well size the magnetic tape and line printer subsystems.

The remainder of this report does not make a distinction between modal, peak, and minimal workloads. The methods discussed apply to all three, and all three should be considered in defining the workload.

There are four major uses of workloads in a system acquisition:

- evaluation of alternative concepts,
- development of preliminary system design,
- selection of a specific proposal, and
- acceptance testing.

The first two take place primarily during the conceptual and validation phase and are discussed in Section III (under Evaluate Alternative Concepts) and in Section IV (under Refine Workload Prediction and Establish Technical Feasibility). The use of workloads during selection is discussed in Section IV (under Evaluate Technical Proposals) and acceptance test workloads are discussed at the end of Section V (under Test System). Nothing more will be added here other than to point out that workloads are used through the full-scale development phase. Hence, a continuing effort to keep the workload definition accurate is required throughout the acquisition process.

PERFORMANCE CHARACTERISTICS

The primary performance characteristic is how responsive the system must be to the users' needs. Secondary aspects are that the system is well utilized and that a means of measuring system performance is provided. These considerations are discussed below.

User Considerations

The primary objective of a computer system is to provide a service to the user, which for performance, means timely outputs. The types of service a user receives can be separated into three different classes: batch (where all the inputs are submitted at once for processing and all the outputs are returned to the user when the processing is completed), interactive (where the user interacts with the system via a terminal, e.g., a transaction processing application or a time sharing system) and real time (where the system directly monitors and/or controls a process without user interaction, e.g., a radar processor). Each of these different classes of jobs has different types of user performance requirements.

There are two classes of batch jobs, each with different performance requirements. Unscheduled batch job (developmental or special requests in support of studies) responsiveness is determined by turnaround time, while scheduled or production job (such as a daily or monthly management summary) performance is based on meeting

the schedule. For interactive applications, the measure of effectiveness is response time, while for real time applications it is the probability of capturing and processing events.

The minimum statement of a performance requirement would be an average, e.g., the average turnaround time should not exceed two hours. For many applications an average does not sufficiently reflect all of the users' requirements. Often a maximum value is also provided, e.g., average response time of five seconds but not to exceed 15 seconds. In some cases even more is stated, e.g., at least 90% of the scheduled batch jobs completed on time, not more than 5% over 15 minutes late, not more than 1% over one hour late and none to exceed two hours late. Each of the three methods of specifying performance requirements (average, average and maximum, and percentile thresholds) apply to turnaround, schedule reliability, and response time requirements. For real time applications probabilities of capturing an event can also be further specified by the addition of more stringent criteria, e.g., 0.1% chance of missing an event and 0.001% chance of missing two consecutive events.

An additional performance consideration is usually associated with interactive and real time applications -- reliability. Mean time between failure, mean time to repair, and maximum allowable down times are some of the typical measures of reliability. These measures, while widely used, do not relate directly to user performance characteristics and it may be more meaningful to express them in terms of the probability of the system being able to process an event and the maximum time period that the system can be down for a given application or transaction. Again this last measure can be specified in more detail, e.g., and average down time of one minute, less than two minutes 90% of the time, and never to exceed five minutes.

Finally, performance requirements will most likely not be the same for all jobs in a given class, i.e., some interactive applications will have more stringent response time requirements than others, and the response time requirements for a given message may vary depending on the situation, e.g., the response time requirements for a message may be relaxed during a degraded mode of operation and the requirements for urgent messages may be greater during peak workload periods, since for C^3 systems peak workloads generally occur during crisis periods.

The major factors that determine performance requirements are operational (age of data, frequency of reporting, urgency and mission requirement). The major technical considerations are

performance requirements versus needs (are the requirements consistent with the operational needs), feasibility (can the requirements be met within the state-of-the-art), cost versus performance (are there less stringent requirements that are still acceptable, but less costly to achieve) and testability (are the performance requirements quantitively stated in terms of parameters that can be measured during system testing). These considerations are all discussed in more detail later in this report.

System Considerations

There are two performance requirements associated with the system: resource utilization and performance instrumentation. Each is briefly discussed below.

Resource Utilization

Saturated resources may cause a bottleneck that affects response time, while underutilized resources may indicate a potential to reduce the overall system cost. The user is normally unconcerned with resource utlization as long as response times are adequate. However, to the system designer and installation manager, resource utilizations are indicators of system efficiency. Feasibility studies during the conceptual and validation phase must consider resource utilizations and system testing should measure resource utilizations to verify that the system profile is reasonable, i.e., the utilizations allow for possible system growth, but are not so low as to indicate overdesign.

Performance Instrumentation

Tools to measure system performance are required for two major reasons: to test the system and to monitor the operational system. The types of tools used to measure performance include accounting packages, software and hardware monitors, and message log tapes. Their use will be discussed in more detail later in this report.

SECTION III

CONCEPTUAL PHASE

During the conceptual phase the mission is analyzed and requirements are documented. Technical, operational, and economic baselines are established and alternative solutions are proposed. These solutions are analyzed using trade-off analyses, experimentation, and other studies to determine viable alternatives.

The conceptual phase performance considerations are divided into two major tasks: determining performance requirements and evaluating alternative concepts.

DETERMINE PERFORMANCE REQUIREMENTS

The performance requirements determined during the conceptual phase can be separated into two areas: workload determination and performance characteristics. While these areas are discussed separately, they are highly interrelated -- performance characteristics are a function of the total workload.

Determine Workload

Normally a C^3 system being developed is an enhancement to an existing system, either manual or automated. In this case an important part of predicting the future workload of the new system is determining the current system's workload. If there is no current system, the future workload can be based on similiar systems' workloads (the frequency of events occurring can be based on a system, either manual or automated, with similiar triggering events and the processing requirements per occurrence can be based on systems with similiar processing requirements.)

Determine Current Workload

There are two different starting points: one where the current workload is manually processed and the other where it is automated. There are also situations where actions are a combination of manual and automated processes; however, such cases can be broken down into separate components that are either manual or automated.

Manual Systems. The major statistic required from manual systems is the number of transactions processed per unit time. Systems involving a significant number of transactions and/or a large number of people, usually have control procedures to log

transactions by shift. Often this raw data is summarized by week, month, or year and historical summaries are available on past frequencies.

For the cases where no data is available on event frequencies, and the computer workload on the new system is anticipated to be significant, sampling techniques, observation, and estimates from the people who process the workload are about the only recourse.

The problem of estimating computer resources required per transaction will be discussed below under Predict Future Workload.

Automated Systems. For systems currently automated there are several means available for determining both frequencies of occurrences of events (batch programs or transactions) and the resources they require. The most commonly available, and normally the starting point, is data from the system accounting package. Accounting data is usually retained, and thus historical data is available for trend analysis. The raw accounting data is not very useful -- it must first be conditioned (incomplete, erroneous or inconsistent records deleted) and then reduced (summarized in reports).

Often some information is not collected in the accounting data, e.g., operating system overhead, file characteristics, and data on interactive or real-time applications. There are several other means of obtaining this type of information. Message logging or transaction tapes provide raw data (requiring conditioning and reduction) on event frequencies and message characteristics, but lacking in resource utilizations. Hardware and software monitors, while not normally providing historical data, are often the only means of obtaining data on operating system overhead and real time application resource utilizations. In considering the use of a hardware or software monitor the incremental value of the data to be obtained must be weighed against the costs of using the monitor.

Predict Future Workload

The workload to be imposed on the system can be derived from the current workload or a similiar system's workload. Event frequencies can be projected using historical data and then, if required, further modified to account for any known changes to the system, e.g., changes in number of terminals or reporting frequencies.

Besides accounting for changes in frequency of event, changes in processing requirements must also be considered. Typical factors

affecting processing requirements of existing applications are changes in data base size and increased accuracy requirements. Changing processing requirements also result from enhancements to existing functions (more information stored on-line), design changes (differing file structures) or new capabilities (automating a manual process). Two techniques are commonly used to predict processing requirements during the conceptual phase:

- estimates based on a similarly designed application are the most commonly used and
- estimates based on the preliminary design concept using modeling techniques when the first technique does not apply.

<u>Performance Characteristics</u>

During the conceptual phase the types of performance characteristics discussed in Section II are determined for the system. Those requirements are for the most part determined by operational requirements. During latter development phases the system level performance characteristics are further allocated to individual subsystems and components within these subsystems. The major performance objective during the conceptual phase is to determine the cost and feasibility of meeting the system level performance characteristics. The commonly used methods of doing this are discussed in the next subsection.

Also during the conceptual phase, performance requirements for measurement tools and conditioning/reduction packages required by the operational system should be identified. Any development efforts in these areas should be integrated into the overall project schedule. The operational instrumentation should be used to the greatest extent possible during testing. It should be identified as a time critical item that must be validated prior to system testing and preferably sooner (so that it can be used to time individual routines, to verify previous estimates, or to identify possible problem areas early in the full-scale development phase). Also, performance charactistics that cannot be measured using the operational instrumentation will require special test instrumentation to be developed.

EVALUATE ALTERNATIVE CONCEPTS

The second major technical consideration (the first being to determine performance requirements) that occurs during the conceptual phase is to evaluate alternative system concepts. The alternatives under consideration at this stage are at a high level

of detail, e.g., centralized versus decentralized, fully automated versus interactive, network architectures and data base locations.

Performance Objectives

There are two types of performance studies that are done during the conceptual phase: preliminary sizing studies and trade-off studies.

The objective of preliminary sizing studies is to estimate, for each of the alternative concepts, the system characteristics (memory access time, memory size, number and speed of disk channels, etc.) of the major system components required to process the workload within the users' performance requirements. Based upon these characteristics, costs can be derived for input to cost/benefits or cost/performance trade-off studies.

Trade-off studies present to the decision maker, for each of several alternatives, data in two or more opposing areas. In the requirements versus need trade-off study the stated user requirements are compared with operational needs, regardless of cost, to assure that they are justified. One of the aspects of the performance versus need study could be to compare the current system's performance with the proposal requirements to verify that the current deficiencies match with the proposed performance improvements. The cost versus performance trade-off studies examine the cost of obtaining the stated performance requirements against less costly systems that may not completely meet all the requirements. The cost versus benefits trade-off study then examines the costs of alternatives in comparison with their anticipated benefits.

All of the technical trade-off considerations have a bearing on determining which alternative is best and the collection of individual studies should be considered as a whole: the requirements versus need versus cost versus benefits study. This study may never be formalized, but the individual components should be consistent and comparable with one another.

Performance Prediction

The objective of performance prediction is to determine how well the alternative conceptual systems meet the performance requirements. The potential uses of several tools/techniques are discussed below.

One means of performance prediction is benchmarking. However, it is infeasible to have vendors configure proposed systems (with highly complex interfaces), develop benchmark workloads (including interactive and real time applications), and instrument the system to measure performance.

For the same reasons (the complexity of C^3 systems and the many alternatives) the development of testbeds or prototypes to predict performance is not practical (from a cost and time stand point) during the conceptual phase. However, if a testbed already exists (e.g., one developed during advanced development) or if there is a system similiar to one of the alternatives, it may be possible to perform limited performance experiments during the conceptual phase.

Predictions based entirely on arithmetic averages of processing requirements and simple workload distributions are not very credible for two reasons: 1) they require many assumptions (lack of queueing, minimal multiprogramming interference, no random arrival rates, etc.), and 2) they have not predicted performance well in those cases where they were applied (largely due to the assumptions required). This does not mean that simple arithmetic models should never be used - they can be the basis for deriving inputs to more complex models during all phases of system development and in the later phases of development can provide credible and accurate predictions for specialized types of subsystems, e.g., a radar signal preprocessor that has only one specific function to perform in a prespecified time period.

The most common method of predicting system performance during the conceptual phase is extrapolation from existing systems that are similiar to the alternative being studied. Such methods can provide predictions that are accurate enough to determine technical feasibility, provided the system chosen has design (both hardware and systems software) and workload similiar to the alternative.

If no similiar system exists, then modeling techniques should be considered. Modeling in the context of this report refers to discrete event simulation, queueing theory, networks of queues and "hybrid" models that are a mixture of discrete simulation and queueing theory. Modeling studies of performance in the conceptual phase are bounded by three considerations:

- a lack of detailed and accurate input data,
- a broad range of alternatives, and

- limited time and resources to apply to the studies.

All three of these points imply that any conceptual phase model can not be very detailed, and the first point implies that the expected accuracy of the predictions will not be great.

SECTION IV

VALIDATION PHASE

During the validation phase, the major technical objective is to further refine and validate the studies and predictions made during the conceptual phase for inclusion into the system specifications and the preliminary development specifications. For performance this includes four areas: refining the workload predictions, allocating the performance requirements, establishing technical feasibility and, if the system will be developed by a contractor, technical evaluation of proposals. These areas are discussed in turn below.

REFINE WORKLOAD PREDICTIONS

The workload predictions derived in the conceptual phase are normally based on extrapolation from current or similiar existing systems. These extrapolations require many assumptions in such areas as:

- increased processing due to requirements for new or enhanced capabilities (such as going from a batch mode to an interactive mode or increased accuracy requirements),
- changes in programming language (assembly to higher order, a different higher order, an optimizing compiler, or another vendors version of the same higher order language) and
- system overheads (operating system, data management system, teleprocessing packages and utility packages).

In order to predict the workload more accurately, the sensitivity of the workload to the assumptions made can be determined. Then, if there are assumptions that have a major impact on the workload, other means can be used to narrow the range of the sensitive parameters. These other means include modeling, the use of prototypes or testbeds and limited benchmarks on existing systems. Which technique is best and when it should be used depends on such factors as technical risk, the time and resources available and which, if any, system components are available.

Those areas that cannot be determined with sufficient accuracy during validation should be identified as areas of high technical risk that will require further study during the full-scale development phase.

ALLOCATE PERFORMANCE REQUIREMENTS

The preliminary system design consists of determining the hardware and software subsystems that will comprise the system. The system performance requirements defined and justified during the conceptual phase are allocated to these subsystems, e.g., a two second response time for a query may be allocated as follows: one-half second for input in the communications subsystem, one second to process the query in the main system, and one-half second for output in the communications subsystem.

The preliminary design/allocation process is iterative. A design is postulated within a given conceptual alternative, the feasibility of this design is evaluated against several criteria (only some of which are performance related), and the design is modified to compensate for any discrepancies. This cycle of design modification/feasibility evaluation is repeated until the design criteria are satisfied for the minimal cost.

ESTABLISH TECHNICAL FEASIBILITY

The exclusive use of either benchmarks or analysis using arithmetic averages is not practical in the validation phase (for the same reasons they were not in the conceptual phase). Arithmetic and statistical analyses are used primarily to interpret the data obtained from limited benchmarks, to predict the new workload based on these analyses, and to reduce the complexity of (sub)system models.

Modeling tools and techniques are the most useful means of establishing technical feasibility. At this point the range of system architectures has been narrowed, more accurate details are available on the workload and system overheads, and preliminary hardware sizing studies have narrowed the range of system components. This additional and more accurate data, coupled with the limited range of possibilities makes possible more detailed simulations and more accurate predictions. These possibilities are also desirable in order to determine detailed hardware characteristics and configurations, to predict with confidence subsystem performance (both user and system) and to identify potential performance risks (highly utilized resources or response times that are marginal or highly sensitive to small changes in either workload or hardware characteristics).

One of the major difficulties of a modeling effort done during the validation phase is validating the model. The actual system does not yet exist and there is no means of comparing model

predictions with an actual system to determine how accurate it is. This was not as great a problem during the conceptual phase where the models were less detailed and the major consideration was relative accuracy rather than absolute accuracy. During validation, it is typical to want model predictions to be within 20% of the actual system. This points out the need for carefully designed experiments to determine the factors to which the model predictions are most sensitive and the use of other means to validate these factors. That is, sensitivity analysis is used to identify high risk areas in the model and these areas are further studied using limited benchmarks, testbed experiments or the development of prototypes (hardware, software or both). Another benefit of using testbeds during validation is to investigate the man-machine interface. Such tests of the man-machine interface are useful for validating response time requirements. This is especially true for manual or batch applications that are being converted to interactive applications.

EVALUATE TECHNICAL PROPOSALS

Proposal evaluation is based on several criteria, e.g., costs, project management plan, proposed schedule and technical considerations. Performance is an important aspect of the technical evaluation of a proposal.

For relatively small, off-the-shelf systems, benchmarks are a good means of determining system performance characteristics. However, most C^3 systems have reached a level of complexity that normally require extensive software (and often hardware) development. This means that system benchmarks prior to selection are impractical. Because C^3 systems are also usually one-of-a-kind systems (or at most a small number of systems) it is normally economically infeasible to have two parallel development efforts. In lieu of a system benchmark, limited benchmarks of specific subsystems that have been identified as high technical risk areas could be used. While not assuring the performance of the overall system, such benchmarks would provide confidence in those areas that are expected to have the greatest impact on system performance.

The bidders often use models that they have developed to predict performance and include these predictions in their proposals. To fully evaluate these predictions would require an extensive period of familiarization and evaluation of the model, which is also impractical.

One possible course of action is to require the bidders to use a specific model, provided as part of the request for proposal, for

their performance predictions. Alternatively, the bidders could be required to provide specific system parameters in their proposals which would be input to a model used by the evaluation team. Both these alternatives require futher investigation to determine the practicality of developing a model that could adapt to a wide range of design alternatives and provide accurate predictions.

SECTION V

FULL-SCALE DEVELOPMENT PHASE

During the full-scale development phase, performance is considered in three areas: developing the detailed design, reviewing the detailed design, and testing the detailed design. These areas are discussed below.

DEVELOP DETAILED DESIGN

The subsystems developed and validated as part of the preliminary design are further refined in the detailed design. Hardware subsystems are subdivided into individual configuration items (CIs) and software subsystems are further subdivided into computer program configuration items (CPCIs). Among the types of information specified for these CIs and CPCIs (referred to hereafter as (CP)CI) are performance characteristics. These performance characteristics represent a further allocation of the subsystem performance requirements that were allocated during the validation phase.

The same considerations apply to this final allocation of performance requirements from subsystem to (CP)CI as did from system to subsystem. The two main considerations are that analyses are performed to justify the allocation, and that they are testable (i.e., quantified in terms of measurable parameters).

These analyses are again based on the same tools as used previously: modeling, measurements from similiar systems, testbeds, and prototypes. In general, each of these tools can be expected to give more detailed and accurate predictions than the previous one. The hardware prototypes used will be preproduction prototypes (breadboards or brassboards representing the detailed design). Software prototypes will accurately depict the mainline coding and be executed against data bases representative of the operational environment. Testbeds will have evolved toward mock-ups of the actual system. Models will be more detailed and many of the inputs that were previously based on assumptions can now be based on prototype or testbed results.

The modeling efforts will, due to the greater level of detailed design data available and the requirement for more accurate predictions (within 5%-10% of actual performance), be based much more on simulation than queueing theory. It may be possible to add this additional detail and complexity to models developed during the

validation phase, but this will often be more costly than developing a new model tailored to a specific performance question and incorporating results obtained from previous models, prototype tests and testbed results.

As models become more detailed, are based on fewer assumptions, and thus closer to representing the final system and more accurate in predicting performance, their value as a tool for use in tuning the operational system increases. For this reason any model being developed by a contractor should be closely monitored by the SPO for operational value, and if it is determined that it is of potential value, the SPO and user should become familiar with the model and request that the contractor provide documentation and/or training on the models use. Such requests will normally incur an additional cost and thus acquisition plans should include the requirement for funds to obtain model training and documentation if desired.

REVIEW DETAILED DESIGN

For (sub)systems being developed by a contractor two formal reviews of the detailed design are conducted: the Preliminary Design Review (PDR) and the Critical Design Review (CDR). (Sub)systems developed by the Air Force, while they do not formally undergo PDRs and CDRs, are subject to the same types of performance considerations that are discussed below.

Preliminary Design Review

After the preliminary design for a (CP)CI has been completed, a PDR is held to review the progress, consistency and technical adequacy of the design approach. Thus, PDRs are not primarily intended to review performance requirements. However, performance is the subject of at least one and possibly two areas that are reviewed. Test procedures for (CP)CIs are reviewed. This includes performance testing (i.e., a performance test case should be identified along with means of measurement). Secondly, any (CP)CIs that are components of a performance critical subsystem (as identified during the validation phase) should have been analyzed to estimate performance. The more critical a (CP)CI, the greater the requirement is to have accurate predictions. In general, less accuracy is associated with predictions based on mathematical analyses than ones based on modeling, testbed results, or prototypes (in that order).

Critical Design Review

There are three areas where performance is considered at a CDR. The results of preliminary qualification tests (PQTs), if available, are reviewed (see below). The system test procedures are reviewed to assure that the system performance requirements will be measured. And, finally, estimates of the overall system loading are reviewed. These estimates are based on the best sources of data available: PQT results, prototype tests, testbed results and modeling. To accurately assess the validity of these estimates, the reviewers should be familiar with the tools used and have sufficient time to examine the details of the analyses prior to the CDR.

TEST SYSTEM

The final and most accurate means of assuring that system performance requirements will be met is through testing. There are three basic requirements for a test: a workload to exercise the system, instrumentation to measure the system, and requirements with which to compare the test results. Workloads have been discussed in Section II. Two important considerations for test workloads are the need for drivers [3] to emulate interactive and real time applications and the possible need to simulate interfaces or situations that may be impossible to actually duplicate, e.g., the interface to an existing operational network or radar signals generated by hostile intruders. The construction of such workloads, drivers and simulated interfaces are difficult and time consuming tasks which must be identified and initiated well in advance of testing.

The requirement for instrumentation tools during testing has also been previously discussed. These tools should also be tested. A previously calibrated workload run in a single thread environment is recommended as the first step in instrumentation validation. Then the complexity of the workload can be gradually increased. Hardware monitors are a useful means of testing software monitors and accounting packages. Comparisons between the outputs can verify the accuracy of and determine the overhead caused by software instrumentation.

The validation of system performance requirements requires workloads representative of the various loads the system will be operating under (modal, peak, and minimal) and the various system configurations (normal, back-up, degraded). Because the interactions between subsystems can affect system performance, performance testing should be based on system wide tests and not individual (CP)CI tests.

The two types of development tests are discussed below.

Preliminary Qualification Test (PQT)

Not all (CP)CIs undergo PQT - only those that are performance or time critical. Both categories include performance considerations. The first category is composed of the high technical risk (CP)CIs identified previously (by using prototypes, testbeds or modeling). The second category includes the performance instrumentation that will be used during system testing.

Formal Qualification Test (FQT)

The system FQT performance objectives are to demonstrate that the system performance requirements have been met and that the performance tools, such as accounting packages and software monitors are accurate. Some of the performance tools may also serve as test instrumentation and therefore have been partially tested during the PQT. One class of tools, that are not instrumentation, which may require testing during the system FQT is any models (simulation, queueing or "hybrid") delivered for operational use. The system FQT provides the first opportunity to validate a system performance model against actual system data.

SECTION VI

SUMMARY AND CONCLUSIONS

SUMMARY

Table I summarizes the uses of performance tools and techniques during system acquisition. More detail on their use is found in the appropriate section. The distinctions made between primary and secondary use are based on a typical system, for a specific system there will undoubtedly be some deviations from this general case.

CONCLUSIONS

Of the ten types of tools depicted on Table I, accounting data, software and hardware monitors, benchmarks, extrapolation from similiar systems, and system drivers have specific, straightforward applications. Analyses using arithmetic averages play an important secondary role in support of other tools in almost every performance area. The uses of testbed results and prototypes have broad application with primary uses in a few areas.

The tool with the broadest range of applications, most of which are primary, is modeling. Its uses range from being the basis for early system architecture decision based on limited data to detailed design decisions based on very specific and detailed data. The accuracy requirements also increase from the conceptual phase through full-scale development. Due to the changing objectives, input data availability and accuracy, and output accuracy requirements, it is normally difficult to have one generalized model that evolves with the system. The development of specific limited objective models is a more practical approach.

The other aspects of modeling stressed were the need for user involvement and familiarity with any models that have potential uses during the deployment phase and the importance of sensitivity analyses to determine technical risk areas. Once the technical risks have been identified strong consideration should be given to validating these portions of the model using testbeds and/or prototype results.

The broad range of primary applications for modeling should not be interpreted to mean that modeling is a panacea. While modeling techniques have the greatest potential benefit, experience has also shown that they entail risks. Models, especially simulations, can be costly to develop, often require more resources to develop than

planned, and do not always provide accurate and/or useful results. For these reasons project management has sometimes been reluctant to pursue a technical recommendation for simulation and instead directed the use of secondary tools -- choosing the less costly, less accurate, less chance of failure option over the more costly, more accurate and greater risk one. In the author's opinion, while this conservative approach may result in less development cost, it has a greater chance of also resulting in a poorly performing operational system with a greater total system life cycle cost. There should be further study in this area to clarify for project management the overall system life cycle benefits of modeling -- with an end toward providing specific modeling guidelines to assist the manager in making the best decisions in a given acquisition.

Table I

Summary of Tools/Techniques*

Use / Tool	Conceptual Phase			Validation Phase				Full-Scale Development		
	Workload Determination	Performance Requirements	Performance Prediction	Workload Refinement	Performance Allocation	Feasibility Establishment	Proposal Evaluation	Detailed Design	Design Reviews	System Testing
Accounting Data	D	D					i			I
Software Monitors	d	d					i			I
Hardware Monitors	d	d					i			i
Analyses Using Averages		p	p	p	p	p		p	p	
Modeling	p	p	P	P	P	P	p	P	P	
Similar System Extrapolation	P	p	P	p	p	p	p			
Benchmarks (Limited)						P	p			
Testbeds				P	p	p		p	p	
Prototypes				p	p	P	p	P	P	
System Drivers							t			T

* D(d) = Primary (secondary) data source
I(i) = Primary (secondary) instrumentation technique
P(p) = Primary (secondary) prediction technique
T(t) = Primary (secondary) teleprocessing driver

REFERENCES

1. AFR 800-2, Program Management, 16 March 1972 and Change 1, 30 April 1975.

2. AFR 800-14, Management of Computer Resources in Systems, Volume 1, 12 September 1975 and Volume 2, 26 September 1975.

3. James, D.L. and D.W. Lambert, Remote-Terminal Emulator (Design Verification Model) - Introduction and Summary, ESD-TR-74-372, Hanscom AFB, Bedford, Mass., December 1974.